



trainUP

Georgescu Alexandru-Ionut
342C3

CUPRINS

1. Descrierea temei
2. Descrierea bazei de date:
 - i. Diagrama bazei de date
 - ii. Structura tabelelor
 - iii. Descrierea constrângerilor de integritate
 - iv. Descrierea procedurilor și funcțiilor
3. Descrierea aplicației:
 - i. Diagrama de clase
 - ii. Structura claselor
 - iii. Diagrama de stări și fluxul de lucru (workflow) pentru aplicație
 - iv. Prezentarea modului în care se face conexiunea cu baza de date
4. Capturi de ecran pentru interfețe și rapoarte
5. Concluzii;
6. Bibliografie

TrainUp

Introducere

O problemă care apare în mod frecvent în rândul unei companii este cauzată de timpul pe care îl petrece o persoană care dorește să participe la un curs în domeniul în care activează. În momentul de față, cele mai multe firme atribuie angajaților cursuri prin interiorul unui fișier text sau excel, aceasta nefiind o metodă optimă pentru niciuna din părțile implicate în acest proces. Pe de o parte, timpul care se pierde prin această asignare este destul de mare, mai ales când ne confruntăm cu un volum foarte mare de date, iar pe de altă parte informația nu este structurată și organizată, fapt ce cauzează o asimilare mult mai grea a informațiilor.

Diverse statistici arată că pregătirea unor cursuri de o calitate superioară îmbunătățește performanța și satisfacția morală a angajatului și sporește reputația și profiturile companiei.

Soluția propune crearea unei platforme dedicată pentru adăugarea de cursuri specifice fiecărui tip de nevoie a unei companii. Pentru a facilita procesul, un angajat se poate înscrie singur printr-o cerere online, iar pentru fiecare cerere inițiată, team leader-ul angajatului respectiv este informat prin intermediul unui mail. Team leader-ul are opțiunea de a accepta sau a refuza cererea, în caz că respectivul curs nu este relevant pentru poziția angajatului. În caz că cererea a fost aprobată de către team leader, acesta este trimisă mai departe către o persoană desemnată să de ocupe de aceste cereri. Dacă cererea este pentru un curs online, aceasta poate fi acceptată instant, iar dacă este pentru un curs fizic, cererea va fi reținută până când managementul companiei va putea să organizeze un spațiu și mai mulți doritori pentru acest curs.

Dorința aplicației este de a câștiga cât mai mult timp persoanelor în procesarea cererilor cât și o mult mai ușoară dirijare a traficului de cereri.

Descrierea bazei de date

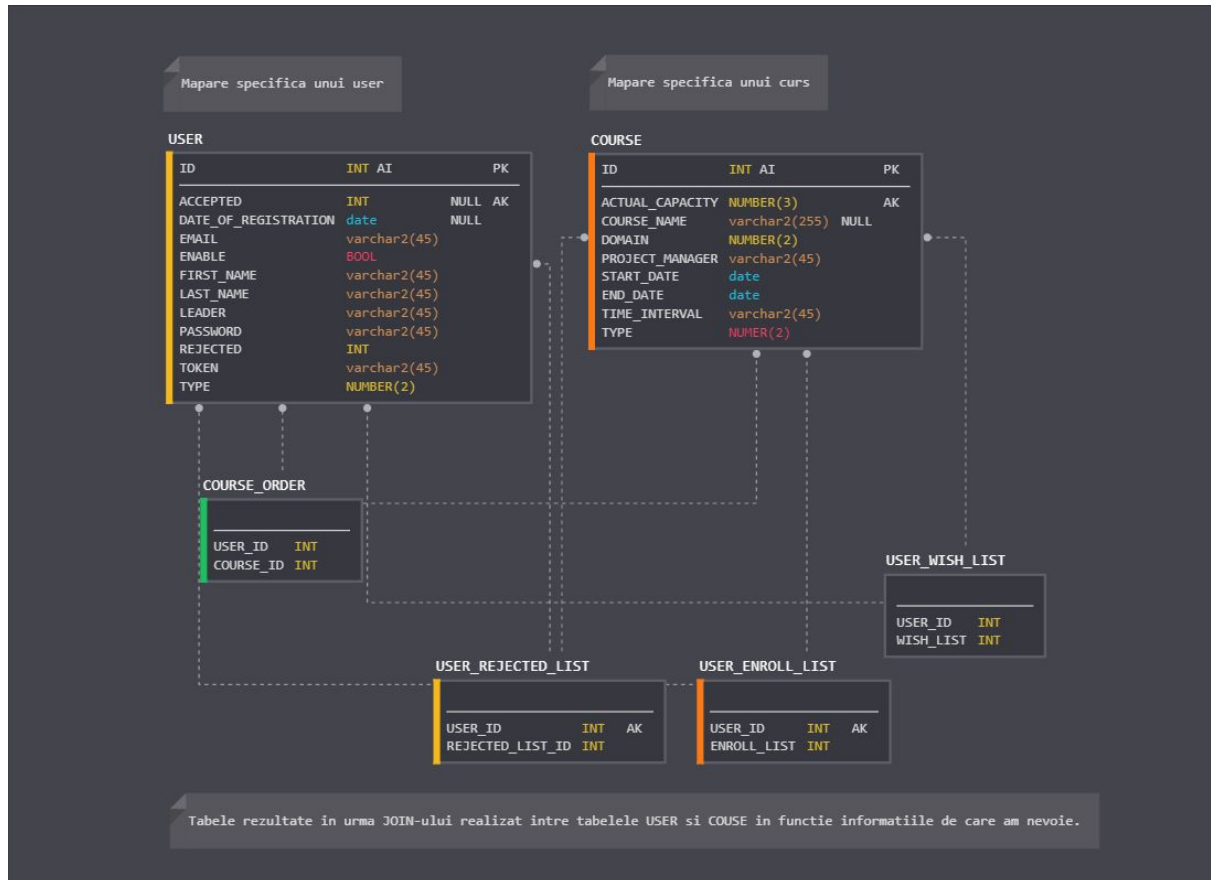


Fig (1) Baza de date

Tabelul USERS

- Reprezinta maparea unui utilizator catre baza de date.
- Contine o cheie prima pe id-ul utilizatorului deoarece se doreste ca fiecare utilizator sa fie unic.
- Numele, prenumele, parola si email-ul sunt datele esentiale pe care utilizatorul le cunoaste si le introduce la intregistrarea s-a pe platoforma.
- Atributele leader, type, date_of_registration, rejected si token sunt atribute esentiale pe care serverul le foloseste in procesarea fiecarui request pe care utilizatorul il cere.

Tabelul COURSES

- Reprezinta maparea cursurilor la baza de date
- Ca si tabelul Users, aceasta contine o cheie primara deoarece fiecare curs e unic, chiar daca un curs se poate desfasura in mai multe intervale de timp diferite, este important ca acest id sa fie unic deoarece vrem sa pastram corectitudinea datelor.
- Acesta contine informatii relevante despre un anumit curs.
- Pentru fiecare curs am avut nevoie sa retin stadiul in care se afla un utilizator la acest curs, de inrolare, asteptare aprobare, respingere sau aprobat, fapt ce cude la nevoia de a implementa inca cateva tabele care se ocupa de acest management de utilizatori.

Tabelele COURSE_ORDER, USER_REJECTED_LIST, USER_ENROLL_LIST si USER_WISH_LIST

- Sunt modele bazate pe id-ul utilizatorului dar si a cursului pentru a putea verifica in ce stadiu se afla un utilizator pentru a putea dirija datele cat mai rapid si in ordine

Tabelul DNS

- Nu am mapat acest tabel deoarece nu este punctul de rezonanta al aplicatiei.
- Acest tabel se ocupa cu maparea adreselor reale al unui utilizator la o adresa virtuala
- Face conversia din nume.prenume@gmail.com sau orice alt tip de serive de mail catre o mapare proprie nume.prenume@trainup.com
- Am folosit aceasta mapare deoarece aplicatia vine si cu un sistem de mail astfel in cat iti poti gestiona gererile direct cu ajutorul telefonului mobil triminand un email catre server iar acesta poate sa te insecie la un curs, poate sa-ti afiseze cursurile la care te poti inscrie sau in functie de timpul de utilizator, sa respingi sau sa accepti cereri.

Proceduri si Functii.

- Am vrut sa fie o aplicatie cat mai reala in cat am introdus urmatoarele lucruri:
 - Atunci cand te inregistrezi trebuie sa-ti confirmi adresa de mail.
 - Atunci cand vrei sa-ti resetezi parola trebuie sa accesezi un link trimis pe adresa ta de email
 - Aici a aparut problema, atunci cand o persoana se inregistreaza, cum stie server ul sa trimita un request pe mail catre persoana respectiva pentru activare sau si mai bine spus, cand cineva vrea sa isi schimbe parola, cum stie serverul catre cine trebuie sa trimita sau cum face diferenta intre toata lumea.
 - Acum apare ideea de TOKEN, am folosit un TOKEN care este folosit pe post de REQUEST ID dar si aici era o problema, cum puteam sa fac astfel in cat acest token sa expire dupa prima utilizarea.
 - Solutia ar fi una foarte eleganta prin folosirea unui trigger care se ocupa de generearea token ului respectiv atunci cand tabela user modificata, atunci cand se insereaza, acesta stie ca trebuie sa introduca un token, atunci cand tabela este modificata pe o anumita intrare(s-a activat enable din tabela pentru a permite userului sa conecteze dupa ce si-a activat adresa de mail sau userul si-a schimbat parola) iar acesta se ocupa de schimbarea token ului facand ca link ul accesat sa expire si fara de folos, asta dupa un request reusit.
- Atunci cand un utilizator executa un request, trebuie sa avem o mapare acestui request la un anumit curs, puteam face acest lucru folosind o procedura care se ocupa cu maparea unui utilizator curent la o lista (o tabela) in care putem sa tinem id ul utilizatorului si id ul cursului iar cum avem doar 4 tipuri de request uri avem clar 4 tabele: una pentru request ul facut de utlizator, a doua pentru requestul facut de catre liderul echipei pentru a aproba un utilizator, a 3 a o reprezinta requestul executat cu succes si utilizatorul aprobat iar a 4 este fix inversul care reprezinta respingerea utilizatorului (am folosit acest tabel deoarece daca un utilizator este respins la un anumit curs, nu vrem ca acesta sa incerce sa faca o cerere catre curs inca o data, project managerul poate oricand sa-l scoata din lista de respinsi si sa-l aprobe si invers).

Descrierea aplicatiei

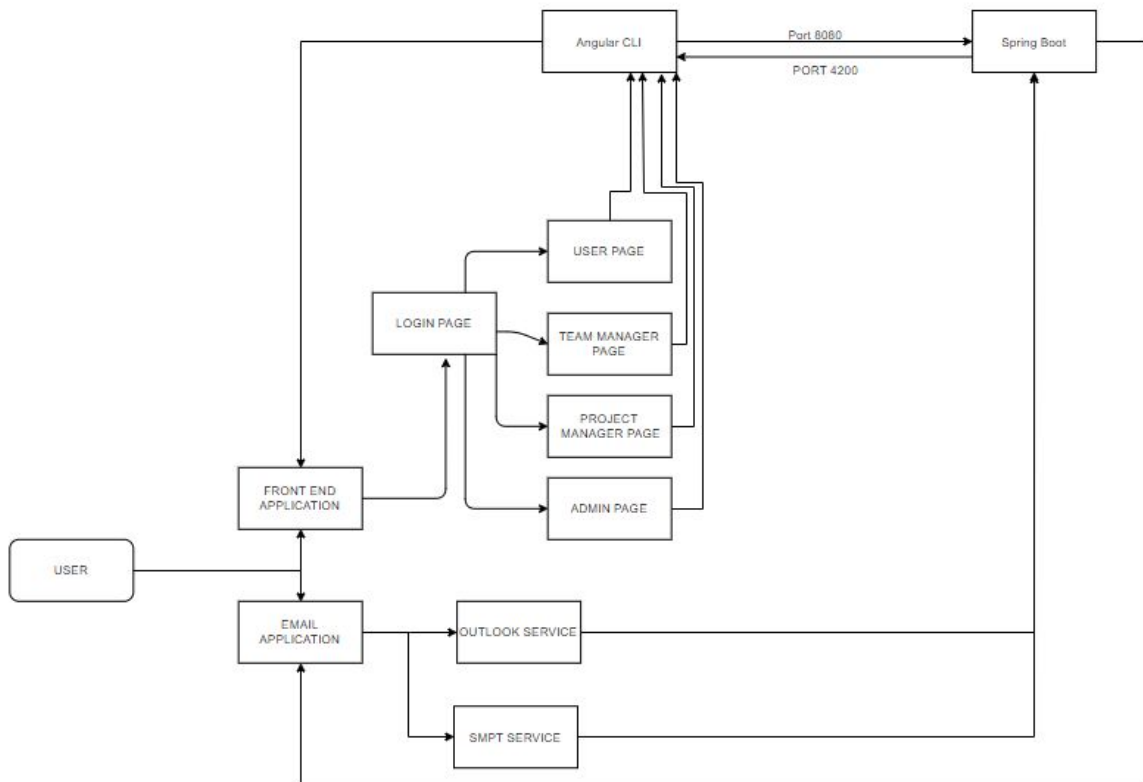


Fig (2) Desciere a aplicatiei

Fluxul de lucru

- In momentul de fata o sa privim aceasta aplicatie din corpul utilizatorului
- Prespuneam ca suntem utilizatorul Alex Georgescu, angajat la firma X care ii are ca si team leader pe Mihai Popescu si project manager pe Andrei Vasile.
- In prima faza Alex trebuie sa decida la ce curs vrea sa se inscrie, poate sa aleaga metoda in care foloseste platforma web sau metoda in care alege mail-ul.
 - Varianta1: Alex alege sa foloseasca platforma web:
 - Acesta intra pe platforma, in data ce s-a locat poate vedea o pagina in care ii sunt afisate cursurile la care acesta este inscris in momentul de fata si inca doua tab uri care sunt reprezentate de cursurile pe care el le-a terminat pana in momentul de fata, respectiv cursurile la care acesta se poate inscrie. Alex alege sa se inscrie la cursul de Baze de Date 2 si apasa pe butonul de request pentru a trimite mai departe cererea catre seful lui de echipa.

- Varianta2: Alex alege sa foloseasca serviciul de mail:
 - Acesta trimite un mail cu subiectul "help" catre adresa de mail a companiei trainUp, trainUpApply@gmail.com.
 - Serverul de backend ii raspunde utilizatorului in timp scurt cu o lista de comenzi pe care acesta le poate folosi, serverul verificand in spate ce tip de utilizator este Alex, acesta ii arata ca poate folosi comanda de "get" (current, past, all) in care acesta poate sa vada informatii despre cursurile pe care le detine sau cursurile la care acesta se poate inscrie. O alta comanda pe care Alex o poate folosi este comanda "add" in care el poate sa se inscrie la un curs.
- In urma acestor doua variante Mihai Popescu primeste o instiintare pe mail in care ii este adresat ca utilizatorul Alex Georgescu doreste sa se inscrie la cursul Baze de Date 2.
- Aceasi poveste se intampla si atunci cand liderul trimite request catre project manager. (liderul poate sa accepte sau sa respinga utilizator si prin intermediul mail ului)
- Project managerul este persoana decisiva, acesta alege daca un utilizator poate sau nu sa participe la acest curs, asa cum acesta este o persoana foarte importanta si are nevoie de cat mai multe informatii, Andrei dispune de o statistica pe care platforma o genereaza pentru a putea vedea cele mai accesate cursuri, cele mai populare luni in care utilizatorii s-au inscris, etc..
- Sa presupunem ca Andrei il accepta pe Alex, in acest moment serverul se ocupa de tot, el vrea in acelasi timp sa-i furnizeze liderului lui Alex ca angajatul lui a fost acceptat dar si lui Alex, in acest scop serverul trimite un mail catre Mihai in care este specificat ca utilizatorul Alex a fost acceptat iar utilizatorului ii este furnizata o invitatie de tip outlook peste care el poate sa apese si se adauga automat in calendarul lui din telefon sau calculator, unde ii sunt specificate locatia, durata si ziua desfasurarii cursului.
- Aceasta este in mare functionalitatea aplicatiei, nu imi ajung 10 pagini sa povestesc in detalii tot ce se intampla in spate, in mare tot ce pot sa mai spun este ca este o platforma foarte bine securizata, orice request se face pe baza de token.

Interfata web

- Mai jos putem sa gasim poze din care se poate observa designul aplicatiei (a fost facut totul manual, nu s-a folosit nici un api care sa aibe un template in spate (mai putin la un pop up verde care se foloseste de material)).



Log in to your account

 E-mail

 Password

Login

You don't have an account? [Register here](#)

Forgot your password? [Reset it here](#)

trainUP

Sort By

Q Search

Add Course

Paul Tiber

Statistics

Log out

Object oriented programming	Check potential participants	Available : 10/20	Starting Date: 2020-09-13	<div>Accept All</div> <div>Reject All</div>																												
Introduction to java	Check potential participants	Available : 20/20	Starting Date: 2019-12-09	<div>Accept All</div> <div>Reject All</div>																												
Linux and unix system administration	Check potential participants	Available : 10/10	Starting Date: 2024-01-13	<div>Accept All</div> <div>Reject All</div>																												
C# and .net	Check potential participants	Available : 10/10	Starting Date: 2020-01-28	<div>Accept All</div> <div>Reject All</div>																												
Basics of C++	Check potential participants	Available : 10/10	Starting Date: 2020-01-03	<div>Accept All</div> <div>Reject All</div>																												
<div>HTML and CSS</div>	Check potential participants	Available : 9/10	Starting Date: 2020-01-12	<div>Accept All</div> <div>Reject All</div>																												
<div>Search by e-mail</div> <table><thead><tr><th>First Name</th><th>Last Name</th><th>E-mail</th><th>Actions</th></tr></thead><tbody><tr><td>Udreanu</td><td>Stoica</td><td>udreanu.stoica@trainup.com</td><td><div>✓</div> <div>✗</div> <div>i</div></td></tr><tr><td>Eda</td><td>Ibram</td><td>eda.ibram@trainup.com</td><td><div>✓</div> <div>✗</div> <div>i</div></td></tr><tr><td>Alex</td><td>Georgescu</td><td>alex.georgescu@trainup.com</td><td><div>✓</div> <div>✗</div> <div>i</div></td></tr><tr><td>Alex</td><td>Bonteanu</td><td>alex.bonteanu@trainup.com</td><td><div>✓</div> <div>✗</div> <div>i</div></td></tr><tr><td>Mihai</td><td>Iamandei</td><td>mihai.iamandei@trainup.com</td><td><div>✓</div> <div>✗</div> <div>i</div></td></tr><tr><td>Vlad</td><td>Manea</td><td>vlad.manea@trainup.com</td><td><div>✓</div> <div>✗</div> <div>i</div></td></tr></tbody></table>					First Name	Last Name	E-mail	Actions	Udreanu	Stoica	udreanu.stoica@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>	Eda	Ibram	eda.ibram@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>	Alex	Georgescu	alex.georgescu@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>	Alex	Bonteanu	alex.bonteanu@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>	Mihai	Iamandei	mihai.iamandei@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>	Vlad	Manea	vlad.manea@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>
First Name	Last Name	E-mail	Actions																													
Udreanu	Stoica	udreanu.stoica@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>																													
Eda	Ibram	eda.ibram@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>																													
Alex	Georgescu	alex.georgescu@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>																													
Alex	Bonteanu	alex.bonteanu@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>																													
Mihai	Iamandei	mihai.iamandei@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>																													
Vlad	Manea	vlad.manea@trainup.com	<div>✓</div> <div>✗</div> <div>i</div>																													
Angular	Check potential participants	Available : 10/10	Starting Date: 2020-01-08	<div>Accept All</div> <div>Reject All</div>																												
Algorithms and Data Structure	Check potential participants	Available : 10/10	Starting Date: 2020-01-06	<div>Accept All</div> <div>Reject All</div>																												

✓

Success

Your request has been sent

Object oriented programming

Check potential participants

Available : 10/10

Starting Date: 2020-09-13

Accept All

Reject All

Introduction to java

Check potential participants

Available : 20/20

Starting Date: 2019-12-09

Accept All

Reject All

Linux and unix system administration

Check potential participants

Available : 10/10

Starting Date: 2024-01-13

Accept All

Reject All

C# and .net

Check potential participants

Available : 10/10

Starting Date: 2020-01-28

Accept All

Reject All

Basics of C++

Check potential participants

Available : 10/10

Starting Date: 2020-01-03

Accept All

Reject All

HTML and CSS

Check potential participants

Available : 9/10

Starting Date: 2020-01-12

Accept All

Reject All

Search by e-mail

First Name

Last Name

E-mail

Actions

Udreanu

Stoica

udreanu.stoica@trainup.com

✓

✗

i

Eda

Ibram

eda.ibram@trainup.com

✓

✗

i

Alex

Georgescu

alex.georgescu@trainup.com

✓

✗

i

Alex

Bonteanu

alex.bonteanu@trainup.com

✓

✗

i

Mihai

Iamandei

mihai.iamandei@trainup.com

✓

✗

i

Vlad

Manea

vlad.manea@trainup.com

✓

✗

i

Angular

Check potential participants

Available : 10/10

Starting Date: 2020-01-08

Accept All

Reject All

Algorithms and Data Structure

Check potential participants

Available : 10/10

Starting Date: 2020-01-06

Accept All

Reject All

Statistics

Courses per type of domain

Courses

0%

25%

37.5%

37.5%

0%

RCA

NFR

GTB

PWCC

Type of domain

Courses with less than 50% enrollment:

Object oriented programming,Introduction to java,Linux and unix system administration ,C# and .net,Basics of C++,HTML and CSS,Angular,Algorithms and Data Structure

Ok

trainUP

Admin Admin

Edit Users

Edit Courses

id	Course Name	Capacity	Actual Capacity	Start Date	End Date	Project Manager	Domain	Time Interval	Delete course
1	Object oriented programming	10	10	2020-09-13	2021-01-13	paul.tiberiu@trainup.com	PWCC	19:00-24:00	
2	Introduction to java	20	20	2019-12-09	2019-12-30	paul.tiberiu@trainup.com	GTB	11:00-14:00	
3	Java Advanced	30	30	2020-01-23	2020-03-13	paul.sava@trainup.com	RCA	11:00-15:00	
4	Linux and unix system administration	10	10	2024-01-13	2025-01-13	paul.tiberiu@trainup.com	NFR	12:00-16:00	
5	C# and .net	10	10	2020-01-28	2020-04-13	paul.tiberiu@trainup.com	PWCC	12:00-16:00	
6	Introduction to agile project management	10	10	2019-11-13	2020-01-06	paul.sava@trainup.com	RCA	19:00-24:00	
7	System architecture	15	1	2020-01-14	2019-12-16	paul.sava@trainup.com	PWCC	11:00-17:00	
8	Welcome to DBTS	15	1	2020-01-16	2020-03-13	paul.popescu@trainup.com	RCA	10:00-16:00	
9	Basics of C++	10	10	2020-01-03	2020-01-04	paul.tiberiu@trainup.com	GTB	10:00-16:00	
10	How to show a presentation	10	10	2020-01-09	2020-01-10	paul.sava@trainup.com	PWCC	10:00-13:00	
11	Waterfall vs Agile	10	10	2020-01-07	2019-12-29	paul.popescu@trainup.com	NFR	13:00-17:00	
12	HTML and CSS	10	10	2020-01-12	2020-01-23	paul.tiberiu@trainup.com	NFR	9:00-16:00	
13	Angular	10	10	2020-01-08	2020-01-28	paul.tiberiu@trainup.com	GTB	9:00-16:00	
14	Algorithms and Data Structure	10	10	2020-01-06	2020-01-18	paul.tiberiu@trainup.com	PWCC	9:00-16:00	
15	German	10	10	2022-01-13	2023-01-13	paul.sava@trainup.com	PWCC	9:00-12:00	

Tehnologii

- Spring Boot (backend) + putin JS pe partea de generare a paginilor de resetare de parola sau generarea de erori.
- Sql
- Angular (frontend) + HTML + CSS

Timpul in care aceasta aplicatie a fost pusa pe picioare

- A fost o aplicatie la care am inceput sa lucrez din septembrie, cu aceasta ocazie am dezvoltat aplicatia la un prag mult mai superior.
- Aplicatia de la primul commit are 4 luni.

Concluzii

1. A fost un proiect destul de greu de implementat deoarece nu pot sa spun ca a fost unul dintre cele mai simple proiecte.
2. Sper ca a fost inteleasa aplicatia cat si functionalizarea acesteia.
3. Chiar daca este un proiect pentru facultate, nu sunt de acord cu utilizarea/expunerea acestei aplicatii fara aprobarea mea.

Bibliografie

<https://spring.io/projects/spring-boot>

<https://spring.io/guides/tutorials/spring-security-and-angular-js/>

<https://www.wikipedia.org/>

<https://angular.io/>

<https://stackoverflow.com/questions/20013672/best-practice-on-generating-reset-password-tokens>

<https://docs.microsoft.com/en-us/outlook/rest/java-tutorial>

Si multe altele..